2012 IEEE RO-MAN: The 21st IEEE International Symposium on
Robot and Human Interactive Communication.
September 9-13, 2012. Paris, France.

Applying Key Typing Pressure to Estimate a User's State of Activity

Takahisa Tani
The Graduate University for Advanced Studies
2-1-2 Hitotsubashi, Chiyoda, Tokyo
101.8430, Japan
tani@nii.ac.jp

Seiji Yamada
National Institute for Informatics
The Graduate University for Advanced Studies
Tokyo Institute of Technology
2-1-2 Hitotsubashi, Chiyoda, Tokyo
101.8430, Japan
seiji@nii.ac.jp

***Abstract*— A user working at his/her desktop computer would benefit from notifications being given at timings that reflect their relevancy to the user's activity and workload. To do so correctly, a notification system should have a way of determining the user's state of activity We propose a novel method to estimate user states with a pressure sensor on a desk. We use a lattice-like pressure sensor sheet and distinguish between two simple user states: busy or idle. The pressure can be measured without the user being aware of it, and changes in the pressure reflect useful information like typing, an arm, the presence of a coffee mug, and so on. We carefully developed features which can be extracted from the sensed data and used a machine learning technique to identify the user state. We conducted experiments evaluating the accuracy of our method and obtained promising results.**

I. INTRODUCTION

In the current office environment connected to the Internet, users tend to get a lot of *notifications*[1] in the form of mails, like in Fig. 1, instant messages, and alerts for application updates. A problem with such notifications is they arrive as they are sent, i.e., without the system being aware of whether the user has time to read them or not. If messages arrive at inopportune times, they can cause stress and reduce the user's productivity [2]. One way of alleviating this problem would be to control the information notification period in accordance with the user's state of activity. In other words, this means a system would need to estimate whether a user's activity can be interrupted or not, and send information only when he/she can be interrupted.

Another approach does not estimate whether the user can be interrupted. A peripheral display [3], [4] is such approach. However, this method of estimating a user's state has other purposes besides notification, e.g., emotional state estimation.

Fig. 1. Notification for arriving e-mails

The system we are interested in would monitor user behaviors like typing, mouse operations, and so on, to estimate whether he or she can be is interrupted or not. There are a number of studies on systems that utilize the frequency of keyboard strokes and mouse operations [5]. However, their methods cannot be applied to a cases in which the frequency does not reflect the user's state of activity or when the user does not use such input equipments. There are also estimation methods that use additional equipments, e.g., web cameras and eyeBlog video glasses, [6], [7], [8]. However, these methods need to monitor user behaviors by taking pictures of their faces and bodies, and thus they could cause psychological stress on the user.

In this study, we developed a novel method to estimate user states by using tabletop pressure. At a desk with a PC, there are changes in pressure on the tabletop caused by the forces of various user behaviors including typing, resting one's arm, lifting a mug of coffee, reading a book, and so on. We considered that useful information for estimating the user's state of activity can be extracted from such slight changes in tabletop pressure. However, there are only a few studies on estimating user states by using tabletop pressure. We hence developed a concrete method for estimating user states by using tabletop pressure. We carefully identified features that significantly contribute to such an estimate. Then, we used the machine learning technique C4.5 to classify user states as idle (i.e., interruptible) or busy (not-interruptible). We conducted preliminary experiments to evaluate the accuracy of our method and obtained promising results.

II. ESTIMATING A USER STATE BY USING TABLETOP PRESSURE

For measuring the tabletop pressure, we spread a pressure sensor sheet having measurement points in a reticular pattern on a tabletop. We assumed that a the user does all his/her work on the sheet and that all objects on the tabletop are placed on it. We investigated the forces involved in typing, resting one's arm, and placing objects on the tabletop, and we found that we need a sensor sheet about 1 meter square with pressure gradation ability of 10 grams.

Fig. 2. LL-Sensor

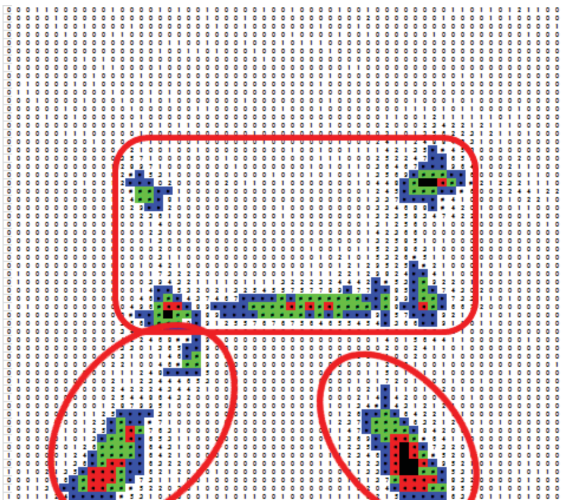

Fig. 3. Output of sensor

Hence, we decided to use the LL-sensor (Xiroku Co., Ltd) in Fig. 2. This sensor leverages the feature of mutual induction. It is 600 millimeters square, and its resolution is 10 millimeters square. It outputs not a physical quantity but a unique value. Figure 3 shows the output of the LL-sensor. The white unit means the lowest value and the blue, green, red, and black means higher values. A keyboard is placed on the square area, and the user's arms are placed on the elliptical areas.

A. Useful features for user state estimation

We pick out feature quantities from the pressure data. In particular, we used key pressing force weight, and location and their changes as feature quantities. We assumed that objects on a tabletop are only a the keyboard of the PC and the users' arms. In the future, we plan to extend this range to include other artifacts like coffee mugs, books, and so on. Note that the weight of a the keyboard is included in the key-typing force.

B. User state estimate

After obtaining features from the raw data, we needed to identify a user state from the data. We utilized classification learning to classify the state into idle (interruptible) or busy (uninterruptible). We used C4.5 as the classification algorithm. C4.5[9], [10] uses a decision tree extended from ID3[11]. We briefly explain how C4.5 works in the following.

Let C be the number of classes and $p(D, j)$ be the proportion of data in D that belong to the jth class. The residual uncertainty about the data to which data in D belongs can be described as

$$Info(D) = -\sum_{j=1}^{C} p(D, j) \times \log_2(p(D, j)) \qquad (1)$$

and the corresponding information gained by a test T with k outcomes as

$$Gain(D, T) = Info(D) - \sum_{i=1}^{k} \frac{|D_i|}{|D|} \times Info(D_i) \qquad (2)$$

The information gained in a test is strongly affected by the number of outcomes and is maximal when there is one data in each subset D_i. On the other hand, the potential information obtained by partitioning a set of data is based on knowing the subset D_i into which a data falls; this $split\ information$,

$$Split(D, T) = -\sum_{i=1}^{k} \frac{|D_i|}{|D|} \times \log_2\left(\frac{|D_i|}{|D|}\right), \qquad (3)$$

tends to increase with the number of outcomes of a test. The gain ratio criterion assesses the desirability of a test as the ratio of its information gain to its split information. The gain ratio of every possible test is determined, and, among those with at least average gain, the split with maximum gain ratio is selected.

In some situations, every possible test splits D into subsets that have the same class distribution. All tests then have zero gain, and C4.5 uses this as an additional stopping condition.

III. EXPERIMENTAL METHOD

A. Experimental environment

We built a simplified desk work environment for eliminating complex factors from this early stage of experimentations. Figure 4 is an overview of it. A participant sits down in front of a desk, and the monitor shows a task window.

Figure 5 shows the window of the main task, and Fig. 6 shows a dialog box asking whether sending notifications at this time is permissible.

B. Features

The task was typing, and the tabletop only had a keyboard (and the user's arms). Hence, we used the following five features.

1) Left foot of keyboard: left foot
2) Right foot of keyboard: right foot
3) Front foot of keyboard: bottom foot
4) Left arm of user: left hand
5) Right arm of user: right hand

These appear as definite areas because the keyboard and the user's arms are the only things on a sheet. We divided

Fig. 4. Overview of the experimental environment

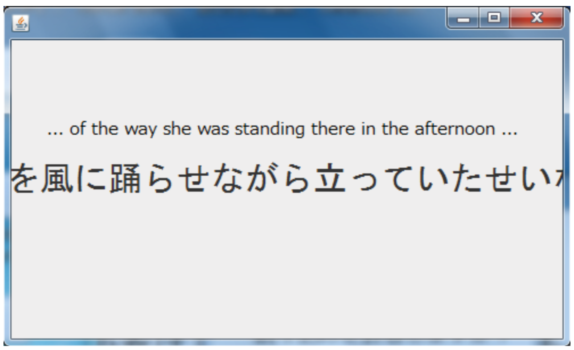

Fig. 5. Main task window

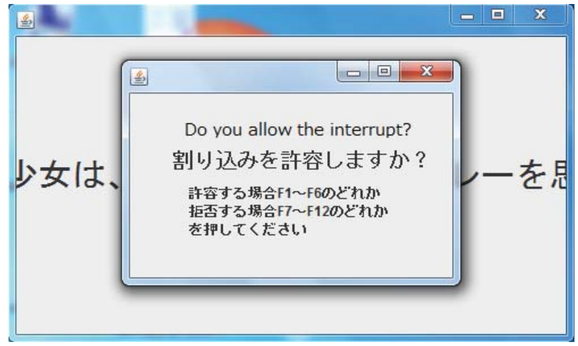

Fig. 6. Notification window.

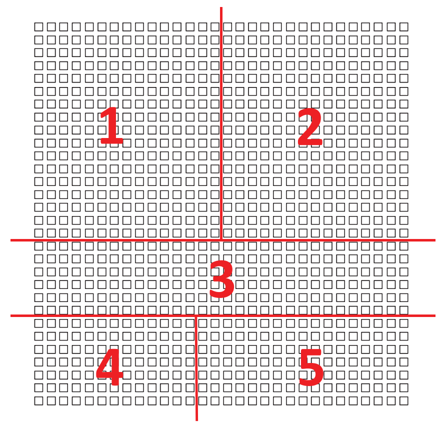

Fig. 7. Regions of LL-sensor.

the sheet into five regions like in Fig. 7 after conducting preliminary experiments.

Next, we chose sensor units which outputted the value over the threshold for each region. We decided to use 20 as threshold after doing some trials. The feature for the time was the average of the chosen units. The training data was the average of the data for 30 frames before the notification, and the user labeled the data with ”allow” or ”reject.”

We used J48 in weka3.6.4[12] as an implementation of C4.5 with two classes (accept, reject) and a confidence factor 0.25.

C. Keyboard Used in Experiments

Since our method uses key-typing force, the estimate of a the user state might be significantly influenced by the properties of the keyboard, e.g. response level, weight, leg shape, and the ground contact area of the foot. Thus, to investigate the influence on the keyboard's properties, we used the following keyboards in the experiments.

- Keyboard A: KFK-EA4XA (Mitsumi Electric Co., Ltd)
- Keyboard B: Realforce 91 NE0100 (Topre Co., Ltd)

Keyboard-A was a standard one. Keyboard-B is one designed for keypunch operators.

D. Participants and Experimental Procedure

The participants were students and staff in the information science department (age; from 23 to 51, mean 35.4, SD = 11.7) and they consisted of seven males and one female. All participants were habituated for key-typing because they worked with PCs everyday.

The experimenter gave the following instructions to the participant:

Instructions to participants

> Please type the scrolling display of characters as correctly as possible. Your typing will be recorded. The scroll speed changes, or stops.
> When typing, the system will ask you whether you can be interrupted or not. Please suppose that the notifications provide you with small amounts of information like weather reports and news. Please push either ”F1 – F6” if you accept it; please push either ”F7 – F12” if you reject it. The window closes after you answer.

The participants used each of the keyboards in turn. The order of keyboard use was counter balanced among participants. Each ”test” involved a participant and a keyboard.

We extracted the features from the output of the sensor and added interruptible data labeled by users. The following

TABLE I
DATA EXAMPLE

left	right	bottom	left hand	right hand	class
26.406	28.761	27.046	30.179	25.327	accept
26.665	28.780	27.270	30.063	26.796	accept
26.559	28.840	27.281	29.860	27.982	accept
26.697	28.952	27.288	29.819	27.816	reject
26.713	28.935	27.142	29.862	26.681	reject
26.843	28.928	27.214	29.911	25.622	reject

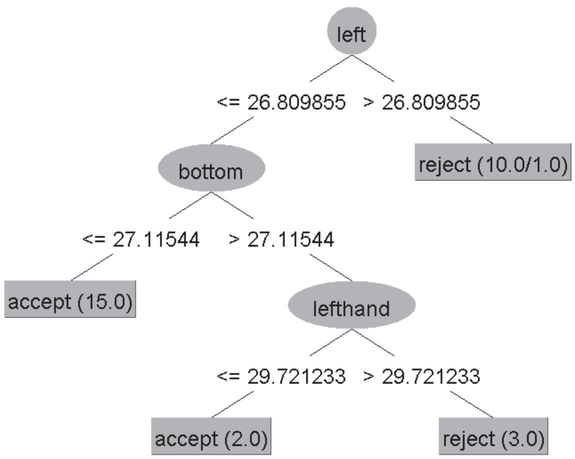

Fig. 8. Decision tree for Participant-C and Keyboard-B

four data sets of the pattern were created:

- A dataset for each test ($n_d = 16, n_t = 30$)
- A dataset for each participant ($n_d = 8, n_t = 60$)
- A dataset for each keyboard ($n_d = 2, n_t = 240$)
- A dataset for total ($n_d = 1, n_t = 480$)

Here, n_d is the number of dataset, and n_t is the number of training data for each dataset.

To remove the influence of the difference in the numbers of accept/reject data, the number of training data was adjusted with ”Resample”, which is one of the training data filters implemented in weka.

The classification was performed with a 10-fold cross validation.

IV. EXPERIMENTAL RESULTS

Table I shows part of the data on a participant who used Keyboard-B, and Fig. 8 shows the decision tree generated from the data.

Figure 9 shows the decision tree made with data of a participant who used both keyboards. The amount of features differs from the first example.

Figure 10 shows the decision tree made with data for Keyboard-B for all participants. Note that this decision tree is much more complicated than the two previous examples. It seems that this was due to the difference in the number of instances used for learning.

The experimental results of 10-fold cross validation are shown in Table II. TP means the rate of correctly classified data, and FP means rate of incorrectly classified data. and F-M. means F-Measure.

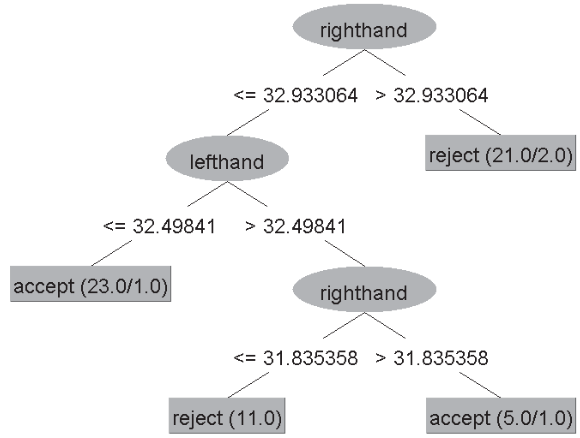

Fig. 9. Decision tree for Participant-B

TABLE II
RESULTS OF 10-FOLD CROSS VALIDATION

Conditions	TP	FP	Prec.	Recall	F-M.
For each test	0.825	0.179	0.837	0.825	0.825
For each participant	0.827	0.175	0.828	0.827	0.827
For each keyboard	0.802	0.195	0.807	0.802	0.802
Total	0.665	0.352	0.670	0.665	0.657

V. DISCUSSION

A. Accuracy of state estimate

The experimental results show our method’s estimation of a user state has about 83% accuracy for each participant and about 80% for each keyboard. We consider this level of accuracy sufficient for preliminary experiments, and it shows our approach of estimating user states with tabletop pressure is promising. In the future, we will introduce more useful features for achieving higher accuracy.

In this study, the experiments were done in an impractical simplified environment. This means there is no guarantee of equivalent accuracy in a real environment. We should therefore increase the accuracy because the experiment acquired information in a restricted environment.

After investigating the decision trees, we did not find any regularity between the sizes of the feature values and theclassifications.

B. Influence of keyboard properties

The learning for uniting data gathered from the two keyboards for each participant and learning for each keyboard for each participant have the same accuracy. In addition, as a result of uniting and learning all the participants’ results for both keyboards, no differences were observed in the results of the two keyboards.

These results mean that states can be estimated without influence from the type of keyboard.

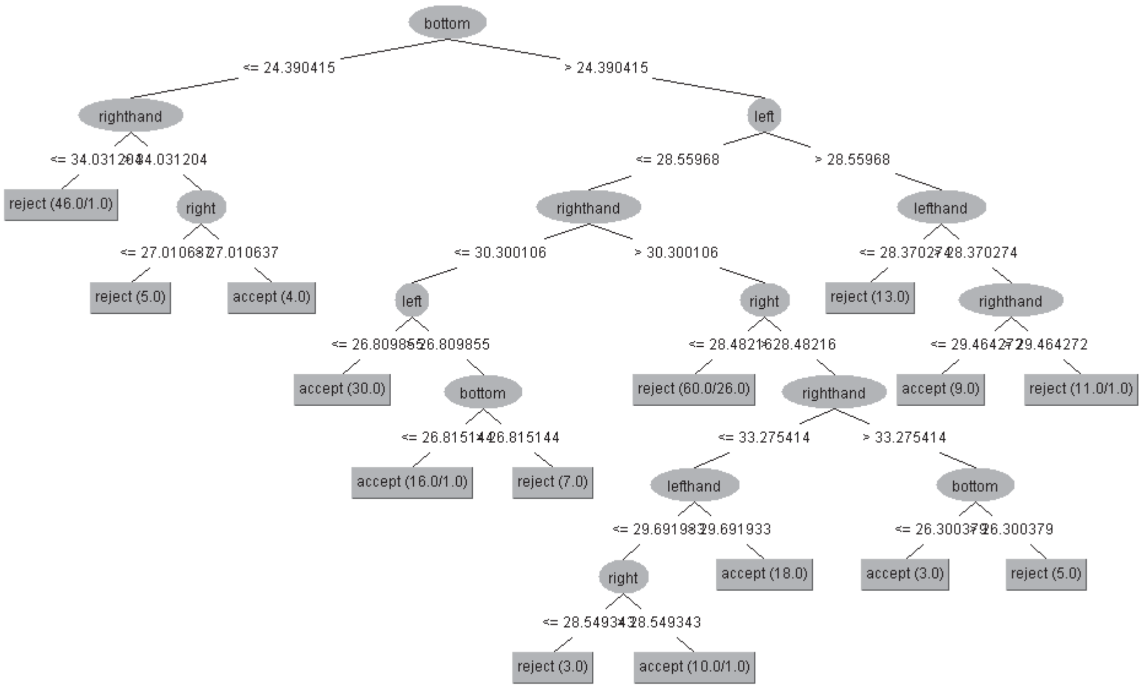

Fig. 10. Decision tree for Keyboard-B

C. Estimation of state of user who is not typing

Our method can estimate the user's state only when the user is doing a keyboard typing task on a desktop because a pressure sensor can not sense any change in pressure when the user is not typing. Thus, in the experimental evaluation, we assumed that a user could be interrupted when s/he wasn't typing.

However, this assumption is not always valid in real environments. For example, a user might be thinking, reading web pages, and watching a movie on the display when they aren't typing, and they would not want to be interrupted in such situations. To cope with this problem, we plan to extend the current features to cover no-typing situations. We will introduce additional features including the pressure of a mug, the shape and area occupied by arms resting on a desktop, which the pressure sensor can sense. We consider these additional features are promising because a user does not pick up a mug frequently and does not change his/her arm position much when concentrating on something.

D. Application of cost-sensitive learning

Cost-sensitive learning can deal with a classification problem in which different misclassification errors incur different penalties [13].

The penalty for misclassification of an interruptible state is relatively lower than that of an uninterruptible state, because the notified information is not urgent and the user tends to reject notifications when he/she is busy. This means we could use a cost-sensitive learning method to classify user states by introducing different penalties for two sorts of misclassifications, and thereby obtain more adequate classifiers for a real environment.

VI. Conclusion

We developed a novel method of user state estimation using tabletop pressure. We conducted an experiment that showed our method could estimate when a user was too busy to receive typical messages.

In particular, the experimental results show the user state estimate was accurate about 83% of the time for each participant and about 80% of the time for each keyboard used. A state estimation independent of keyboard characteristics was also found to be possible.

In the future, we will use richer features taken from real experimental environments. This will help to increase accuracy and make it possible to estimate activity states of users when they are not using the keyboard. For that purpose, we will try to determine the optimal number of the features. In addition, we will assess the utility of cost-sensitive learning.

References

[1] S. T. Iqbal and B. P. Bailey, "Effects of intelligent notification management on users and their tasks," in *Proceedings of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, ser. CHI '08. New York, NY, USA: ACM, 2008, pp. 93–102. [Online]. Available: http://doi.acm.org/10.1145/1357054.1357070

[2] B. P. Bailey, J. A. Konstan, and J. V. Carlis, "The effects of interruptions on task performance, annoyance, and anxiety in the user interface," in *Proceedings INTERACT '01*. IOS Press, 2001, pp. 593–601.

[3] L. J. Williams, "Peripheral target recognition and visual field narrowing in aviators and nonaviators." *Int J Aviat Psychol*, vol. 5, no. 2, pp. 215–32, 1995. [Online]. Available: http://www.biomedsearch.com/nih/Peripheral-target-recognition-visual-field/11540258.html

[4] D. S. McCrickard, R. Catrambone, and J. T. Stasko, "Evaluating animation in the periphery as a mechanism for maintaining awareness," 2001.

[5] C. Epp, M. Lippold, and R. Mandryk, "Identifying emotional states using keystroke dynamics," in *Proceedings of the 2011 Annual Conference on Human Factors in Computing Systems (CHI 2011)*, Vancouver, BC, Canada, 2011, pp. 715–724.

[6] D. Chen, J. Hart, and R. Vertegaal, "Towards a physiological model of user interruptability," in *Proceedings of the 11th IFIP TC 13 international conference on Human-computer interaction - Volume Part II*, ser. INTERACT'07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 439–451. [Online]. Available: http://dl.acm.org/citation.cfm?id=1778331.1778380

[7] J. Fogarty, S. E. Hudson, C. G. Atkeson, D. Avrahami, J. Forlizzi, S. Kiesler, J. C. Lee, and J. Yang, "Predicting human interruptibility with sensors," *ACM Transactions on Computer-Human Interaction*, vol. 12, no. 1, pp. 119–146, 2005.

[8] A. Jaimes, "Posture and activity silhouettes for self-reporting, interruption management, and attentive interfaces," in *Proceedings of the 11th international conference on Intelligent user interfaces*, ser. IUI '06. New York, NY, USA: ACM, 2006, pp. 24–31. [Online]. Available: http://doi.acm.org/10.1145/1111449.1111463

[9] J. R. Quinlan, "Induction of decision trees," *Mach. Learn*, pp. 81–106, 1986.

[10] ——, *C4.5: programs for machine learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993.

[11] J. Quinlan, "Improved use of continuous attributes in c4. 5," *Journal of Artificial Intelligence Research*, vol. 4, pp. 77–90, 1996.

[12] F. E. et. al., "Weka manual for version 3-6-4," 2010.

[13] C. Elkan, "The foundations of cost-sensitive learning," in *In Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, 2001, pp. 973–978.